

Implementation of neighbor communication algorithm using multi-NICs effectively by extended RDMA interfaces

Yoshiyuki Morie^{1,2} and Takeshi Nanri^{1,2}

1. Research Institute for Information Technology, Kyushu university, Japan, 2. CREST, JST, Japan
Email: morie.yoshiyuki.404@m.kyushu-u.ac.jp

Abstract

A neighbor communication algorithm for large messages in K computer was proposed. The key idea of this algorithm was to divide messages into fragments according to the number of neighboring processes and the number of NICs available on each node, so that the bandwidth of each NIC was fully used. To show the effectiveness of the proposed algorithm, an implementation of the algorithm for six neighbors with four NICs by MPI functions was examined. Experimental results showed that, for large messages, the performance of the proposed algorithm was two times faster than the existing one in mpich-3.0.4. However, in medium message size, proposed algorithm is lower performance than existing one. Therefore, the proposed algorithm is implemented by extended RDMA interfaces instead of MPI functions. This implementation became faster than existing one in medium message size too. This result showed the proposed neighbor communication was widely effective.

Neighbor Communication Algorithm

Neighbor communication sends and receives m bytes message to/from the $2r$ neighboring process respectively in process space of r -dimension rectangular parallelepiped. It is assumed that processes allocation on network topology is in agreement in the process allocation in a program. A neighbor communication algorithm for large message sizes is proposed. In this case, it becomes important to utilize the communication bandwidth of all the NIC effectively. On the other hand, the communication algorithm for small message size aims to reduce the number of communications and hops. In this paper, since the process allocation on network topology is the same as the process allocation in a program, the low delay algorithm for small message is easy.

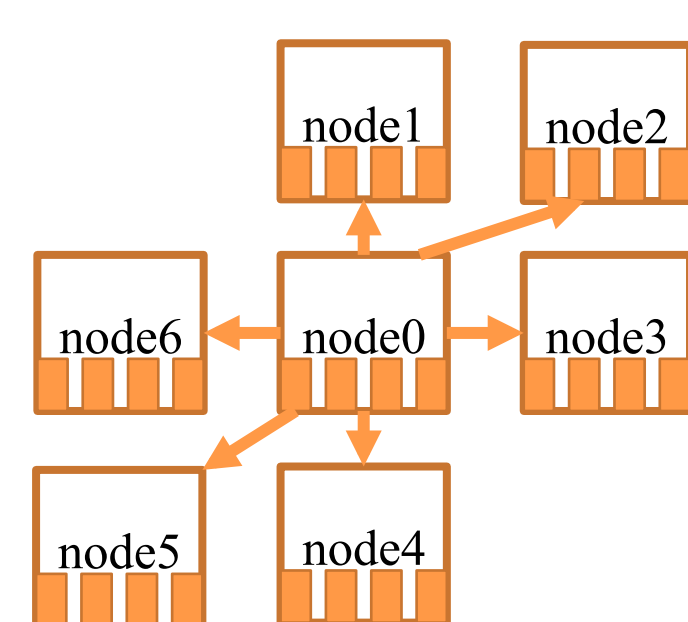


Fig1. Native neighbor communication, 6neighbors and 4-NICs

Authors[1] proposed the algorithm of neighbor communication as follow. If the number of neighboring processes and the number of NIC are in agreement, the communication bandwidth of NICs can be used fully. However, when the number of neighboring processes is larger than the number of NICs, it is possible that the bandwidth is unable to be used sufficiently. If the number of neighboring processes is $2r$ and the number of NICs is n , the communication bandwidth of all the NIC can be used up by transmitting $2r/n$ messages at each NIC. In order to use the communication bandwidth of NIC equally, a $1/n$ message is communicated in $2r$ step. If it is possible to reduce a fraction $2r/n$ by d , it is able to send d/n message as a block. In other words, d/n messages can be sent in $2r/d$ steps.

The model of neighbor communication algorithm for 6-neighbors and 4-NICs

- For example, neighbor communications that has 6-neighbors and 4-NICs is discussed about from now. First, four communications are issued by four NICs, after that two communications are issued by two NICs. Two NICs are not used in the communication of second step. The prediction communication time of native neighbor communication algorithm is shown by a following equation. M is message size on neighbor communication. L and B shows a communication latency and a communication bandwidth respectively.

$$2 \times L + 2 \times \frac{M}{B}$$

- On the other hand, proposed algorithm issues six messages, because the number of neighboring processes is six. Therefore, a communication bandwidth can be used up by transmitting $3/2$ messages per each NIC. As shown in Figure 1, in order to transmit $3/2$ messages by each NIC, data are divided to half, and these data are transmitted in three steps. Then, data transmission is executed so that there are no NICs that are not used in each step. This algorithm is called three steps neighbor communication algorithm. The prediction communication time of proposed neighbor communication algorithm is shown by a following equation.

$$3 \times (L + \frac{3}{2} \times \frac{M}{B})$$

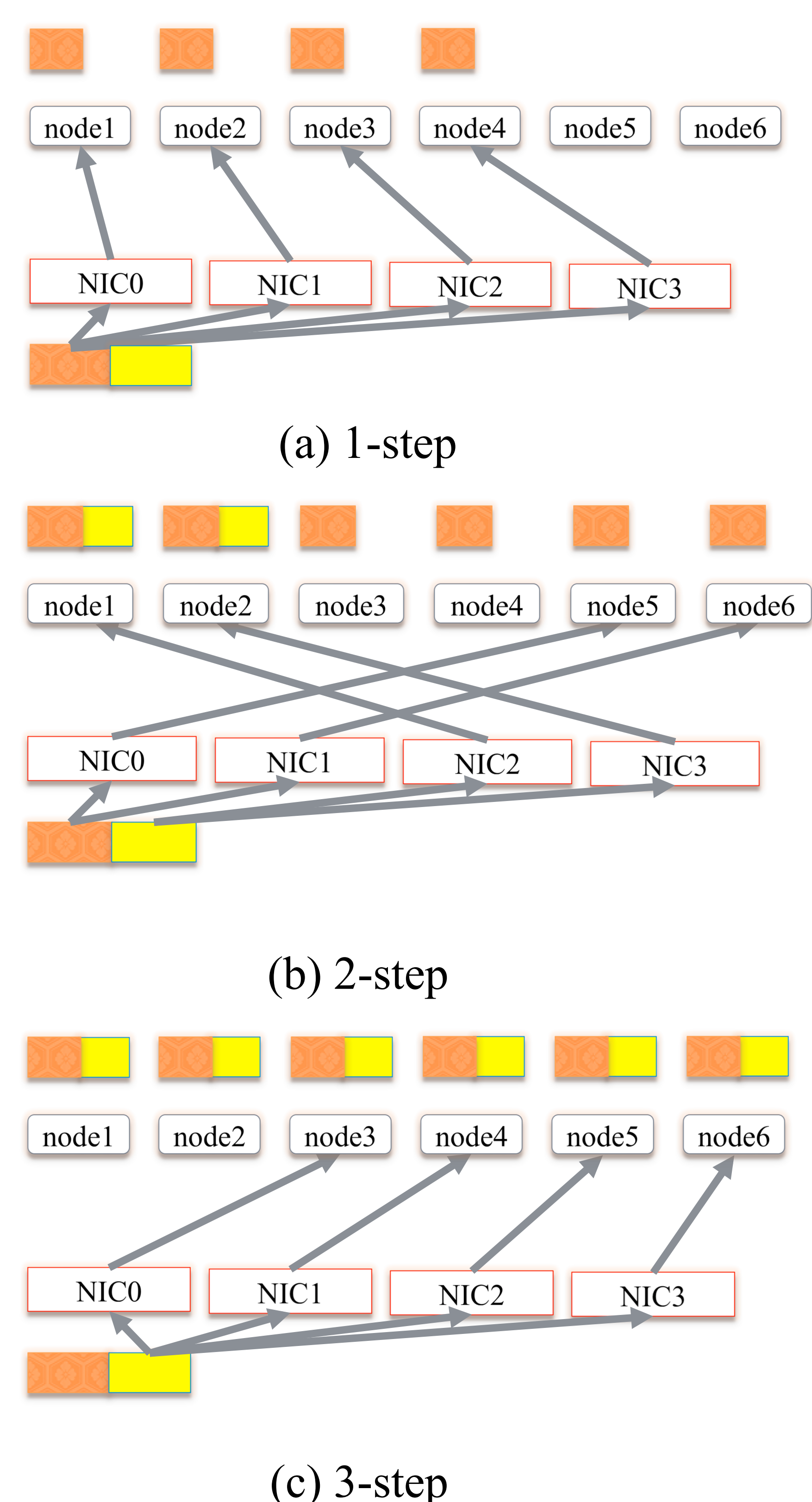


Fig2. 3-step algorithm of neighbor communication

Preliminary evaluation: Neighbor communication by MPI functions

This experiment that compares the performance of the three steps neighbor communication algorithm with the native one is conducted. Both of these algorithms are implemented by MPI_Isend, MPI_Irecv and MPI_Waitall. The preliminary evaluation is executed in FX10 which is family of “K computer” at Tokyo university. Table 1 shows the speculation of FX10.

Table 1 Experimental environment

Machine name	Fujitsu PRIMEHPC FX10
cpu	SPARR64TM Ixfox 1.848GHz (16 cores)
memory	32GB
The number of nodes	4800
MPI library	Fujitsu Technical Computing Suite v1.0
network	5GB/sec
topology	6 dimensional mesh/torus

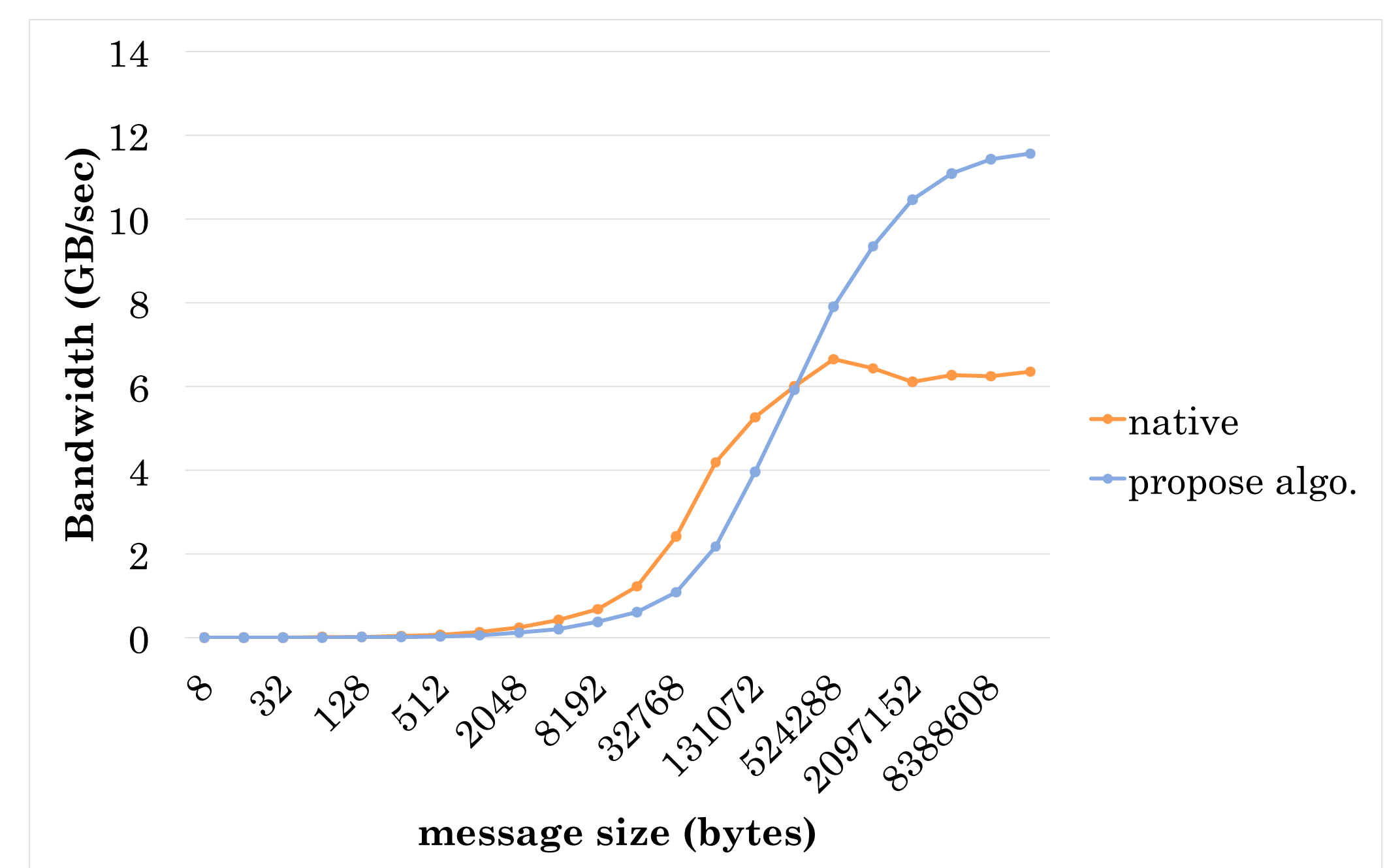


Fig.3 Bandwidth of each neighbor communication

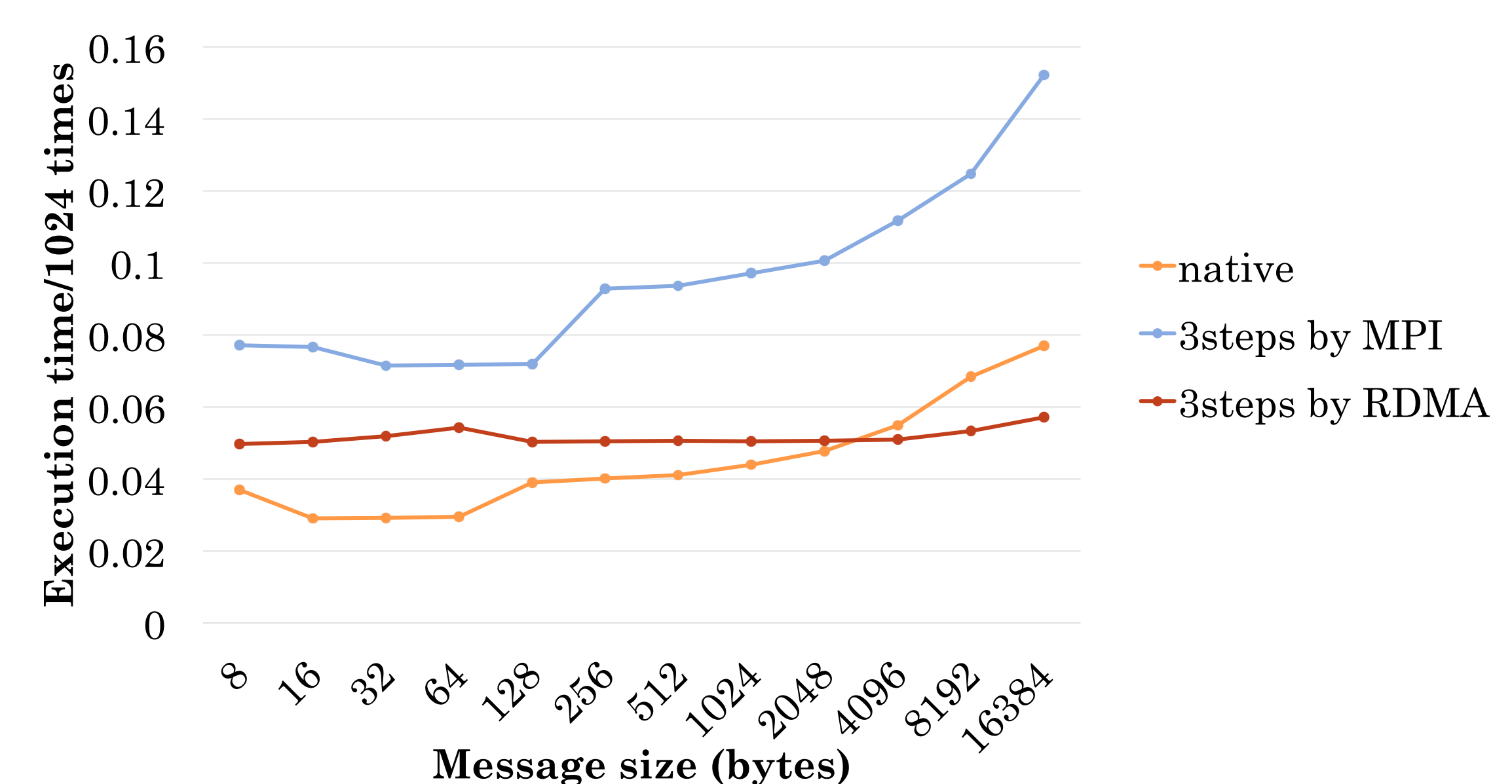
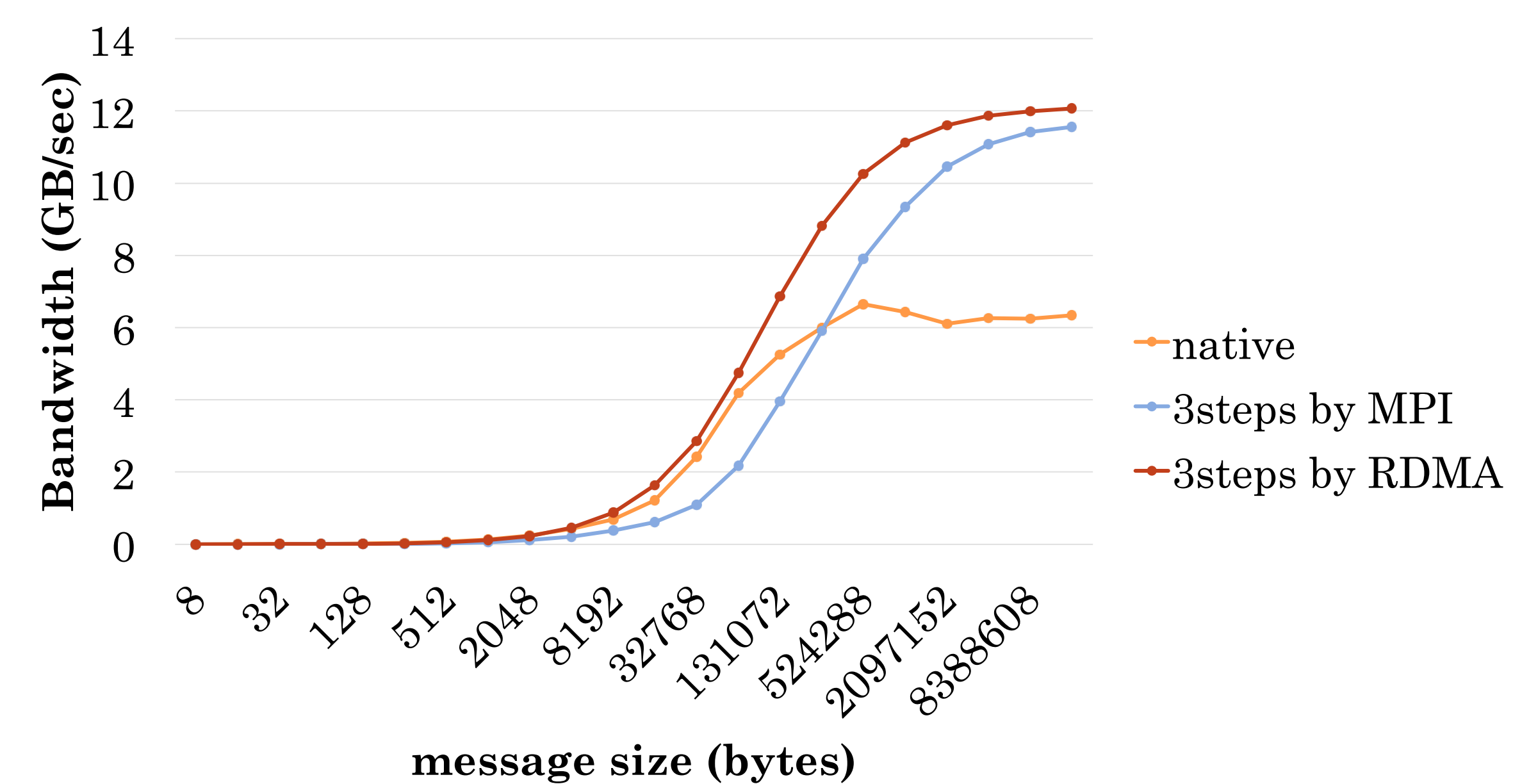
Fig.3 shows the communication bandwidth of each neighbor communication. “native” shows the existing algorithm and “3steps” shows the proposed algorithm. From this graph, the proposed algorithm from 512KB becomes faster than the existing one. For example, with 16 MB, the bandwidth of the proposed algorithm is 1.9 times higher than that of exiting algorithm.

Evaluation experiment: Neighbor communication by extended RDMA interfaces

In the previous experiment, the proposed algorithm until 512KB becomes later than the existing one. Because of this, the proposed algorithm by MPI functions can not use widely. Therefore, the proposed algorithm is implemented by extended RDMA interfaces by Fujitsu Ltd. These interfaces reduce MPI layer software overhead. Moreover, if same buffer is used when neighbor communication executes, reduce the memory registration and exchanging remote base address. Table 2 shows extended RDMA interfaces

Table 2 Extended RDMA interfaces

API	Functionality
FJMPI_Rdma_init	Initialization of extended RDMA interface
FJMPI_Rdma_finalize	Finalization of extended RDMA interface
FJMPI_Rdma_reg_mem	Memory registration
FJMPI_Rdma_get_remote_addr	Get remote DMA address
FJMPI_Rdma_put	RDMA Write communication
FJMPI_Rdma_get	RDMA Read communication
FJMPI_Rdma_poll_cq	Check RDMA completion



Conclusions and future works

- Neighbor communication implementation using extended RDMA interfaces was proposed for FX10. The performance of this implementation improves for medium and large messages, and it was able to realize twice the performance at a maximum to the existing one.
- Future work is consisting of the following. The neighbor communications of other communication pattern are developed. Furthermore, the neighbor communication algorithm when communication contentions occur is developed.

Acknowledgement

This research was conducted using the Fujitsu PRIMEHPC FX10 System (Oakleaf-FX) in the Information Technology Center, The University of Tokyo.

References

- Y. Morie and T. Nanri, “A neighbor communication algorithm with making an effective use of NICs on multidimensional-mesh/torus”, International Conference on Simulation Technology, Sep. 2013.